Documentation for scatter_field_maxwell.py

This Python file simulates electromagnetic wave propagation using the finite-difference time-domain (FDTD) method. It visualizes the electric field over time, specifically focusing on the behavior of the field in a defined grid.

Functions Overview

The code defines several functions for interpolating auxiliary fields and then simulates the propagation of electromagnetic waves through a grid.

Function: Hx_aux_interp

Purpose: Performs interpolation for the Hx auxiliary field.

Parameters:

- `i0`, `j0`: The current grid indices for interpolation.
- `i_start`, `i_end`: The starting and ending indices for the row dimension.
- `j_start`, `j_end`: The starting and ending indices for the column dimension.

Returns: The interpolated value for the Hx auxiliary field.

Example: If `i0 = 5`, `j0 = 5`, `i_start = 4`, `i_end = 6`, `j_start = 4`, and `j_end = 6`, the function will return the interpolated Hx value based on these indices.

Function: Hy_aux_interp

Purpose: Performs interpolation for the Hy auxiliary field.

Parameters: Same as `Hx_aux_interp`.

Returns: The interpolated value for the Hy auxiliary field.

Example: Similar to the previous function, this will return the Hy value based on the specified grid indices.

Function: E_aux_interp

Purpose: Performs interpolation for the E auxiliary field.

Parameters: Same as `Hx_aux_interp`.

Returns: The interpolated value for the E auxiliary field.

Example: Using similar parameters, this function calculates the E field interpolation value.

Global Variables Overview

The following variables are initialized to define the simulation parameters and grid sizes:

- `num_timesteps`: Total number of simulation time steps, set to 1200.
- `n`: Size of the grid (1000).
- `m`: Another dimension of the grid (400).
- `epsilon0`, `mew0`: Constants representing permittivity and permeability of free space.
- `Ez`, `Hy`, `Hx`: Arrays initialized to hold field values.

- `E_aux`, `H_aux`: Arrays for auxiliary field storage.
- `c`: Calculated speed of light in the medium.

Simulation Loop Overview

The main loop runs for a specified number of time steps. In each iteration, it updates the field values based on the previously calculated values, applying various boundary conditions and interpolations. The electric field `Ez` is visualized using Matplotlib at specific intervals.

Visualization

The simulation uses `matplotlib` to create an animated visualization of the electric field over time. Each frame of the animation is generated based on the current state of the field.

Note: This documentation is intended to provide a friendly introduction to the code. For further details, feel free to reach out or refer to additional resources on electromagnetic simulations.